DOCUMENT RESUME

ED 250 697 CS 208 653

AUTHOR Janello, Pam
TITLE Software Evaluation for the Teacher of the English Language Arts. ERIC Digest.
INSTITUTION ERIC Clearinghouse on Reading and Communication Skills, Urbana, Ill.
SPONS AGENCY National Inst. of Education (ED), Washington, DC.
PUB DATE 84
CONTRACT 400-83-0025
NOTE 9p.; Provided in both typewritten version and one-page typeset version.
PUB TYPE Information Analyses - ERIC Information Analysis Products (071)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Computer Assisted Instruction; *Computer Software; Elementary Secondary Education; *English Instruction; *Evaluation Methods; Instructional Materials; *Language Arts
IDENTIFIERS ERIC Digests; *Software Evaluation; Theory Practice Relationship

ABSTRACT

Intended for administrators and policy makers as well as teachers, this digest offers selection strategies for consumers of English/language arts computer software. Following the introduction, the digest discusses initial considerations when evaluating software, including examining both the accompanying documentation and the actual program. It then outlines what to look for when first viewing the program in terms of organization and structure and the quality of the program's feedback for the user's answers. Next, the digest covers the pedagogical issues to consider during the first viewing. It then offers criteria for a second viewing, in which the evaluator intentionally makes errors that a student might make to examine how the program responds to confused or slow learners. Finally, the digest offers suggestions for dealing with software publishers. (HTH)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
X This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

ED250697

ERIC Digest

SOFTWARE EVALUATION FOR THE TEACHER OF THE ENGLISH LANGUAGE ARTS

by

Pam Janello

Sifting good software from bad is not difficult, but it is not like selecting textbooks. If you use the same strategies, you may waste both time and money. You can flip through many textbooks and get a good idea of their worth because you are sophisticated about what you are looking for. Also, when teaching with textbooks, you are still a major part of the show--preparing, emphasizing, explaining, highlighting, filling in the gaps, making up for weaknesses. The book will not slam shut if a student forgets to turn the pages in a certain manner. A computer program, however, can create frustration and confusion for the student. It can, in effect, "break down" if it is not thoughtfully designed. Therefore, teachers must learn selection strategies in order to become sophisticated consumers.

What are the initial considerations?

Never buy software without a thorough examination. You should look at two parts of the software: the documentation and the actual, running program. The documentation is the written explanation that comes with the computer disk. Beware of any software that does not have such documentation, as lack of documentation usually means the person who produced the program didn't think it through or didn't bother with explanatory material.

The documentation should help answer some basic questions. Are the stated objectives of the program something you need for your students? What equipment specifications does the program take? If the program needs 64,000 units of memory and your computers have only 48,000, do you want to buy additional memory chips for all your machines so that this program will run? Does the success of the program depend on color--if so, do you have color monitors? Trying to

CS 208 653

distinguish the sentences written in red from the ones in blue on a black and white screen is tiresome, if not impossible. Does the program need disk drives or cassette players? If disk drives, how many? Is one of the strengths of the program that it will print out a paper copy of the student's work? If it is and you either don't have access to a printer or have one printer for thirty students, maybe the program isn't for you.

When first viewing the program, what should you look for concerning organization and structure?

Suppose you like the subject matter, accept the stated objectives, and have the equipment. It's time to look at the program for the first run-through--and this ought to be the first of several if you are going to be a competent consumer. Concentrate on three areas during the first run: Does the program carry out its stated objectives? Is it user-friendly? Does it have sound pedagogical structure?

You probably do not need help deciding if the program carries out its stated objectives, because that's a regular part of teaching. But what about user friendliness? Is the program easy to use? Are there clear directions, or do you need technical knowledge about computers to make this program work?

Examine the program for ease of operation. If you have to press the Return key after your input, does it tell you to do it? Is the program menu driven? That is, do you start out with a page of choices, or do you have to start in the same place every time--reading directions at the beginning? What about a way to quit if you get bogged down or must leave? Can you get out of the program short of pulling the plug? Can you go back to see those directions you thought you didn't need in the first place, or thought that you'd remember after a quick read through?

What should I know about feedback?

It's important to look at both positive and negative feedback. These terms do not mean good and bad. Rather, they refer to the feedback given for correct and incorrect responses. Is the feedback appropriate? Too much positive feedback isn't useful. Some programs go too far, using the student's name and a string of superlatives to reward a correct response. "What a marvelous job, Charlie. You sure know your stuff." This approach is boring and often condescending. Field studies show that a simple OKAY or CORRECT is often enough.

Does the negative feedback help point students in the right direction? Computers can handle many possible answers in many forms in a helpful way--if the programming was done by a competent programmer and a designer who knows the content. Feedback that says "THINK!" to students who thought they were thinking but missed the answer anyway is useless. A more helpful feedback design is the corrective feedback paradigm (CFP). This is a system that reruns missed questions through a questioning sequence at specified intervals to reinforce retention.

Finally, there are some programs that won't let students out until they have answered five or ten or fifteen questions correctly in a row. Some students just can't do that, and they get so frustrated by a lack of success that they quit. Keep this in mind as you evaluate a program. In addition, remember to check whether the program keeps track of the student's work, giving some type of summary and evaluation at the end.

<u>What pedagogical issues should I consider during the first viewing?</u>

There are several important issues. For example, is the sequencing pedagogically logical? Does the program permit the student to interact--think, respond, wonder, predict--or does the student just read an electronic workbook? (You can buy quite a few workbooks for the cost of one piece of software, much less the computer.) The more interactive the program is the better. Get students involved in this

evaluation and make sure that involvement is worthwhile for both their education and their time.

Look carefully at the language. Is it too formal, or too laden with slang? Is it so dependent on fad that it will be outdated in two years? Finally, do graphics and/or sound add anything to the program, or are they window dressing?

What should I look for in a second viewing?

If the program seems to pass the test so far--you like the objectives, it's user friendly, and the pedagogy is good--it's time for a second round of assessment, in which you consciously make errors. This may seem hard. Teachers like to do things correctly and to be neat. On this run-through, however, you'll need to force yourself to be a confused and/or slow learner.

Find out what happens if you just hit the return key with no input. Can you do it three times, get the correct answer, and page on through or do you have to make an attempt to answer? What does your program do with perceptually correct but literally incorrect answers--especially misspellings or parts of names, abbreviations? What does the program do with a totally off-the-wall response--treat it as such or say, "Wow, that's interesting, Johnnie." What happens if you hit the escape key, hit many keys simultaneously, or randomly tap the keyboard? Try it. If the program is capable of handling such things, it's said to be "bullet proof." If the program isn't designed to handle accidental or intentional problems, it's going to backfire on some students.

If you look at these aspects, you will have given the program a good preview. If you still like the program at this point, then it's time to turn the program over to some of your students. Choose at least a couple--quick/bright and slow/unmotivated and get their opinions. Their actions and reactions will reveal much more about the programs.

What should I know about dealing with software publishers?

If publishers will not let you inspect programs on a trial basis, beware. They are not entirely at fault, however. Software development, especially well-done material, costs money. If they send out a disk for preview and you copy it and send it back, they stand to lose a great deal of money. So allow for the publisher's qualms and make arrangements for previewing. Send a letter guaranteeing that you will not copy illegally. In addition, request to see the real program rather than shortened preview disks. If they won't send just the disks, ask them to send a sales representative to do a demonstration and then try to get time for your own evaluation. Sometimes several schools can join together and invite several publishers to put on a software fair with a wider range of materials. It takes planning, but it is a way to see the material.

Another factor to check on is how to get a backup disk or multiple copies. You should be given either a backup disk or provision to make your own. Also, how does the publisher deal with your need for five, ten, fifteen or thirty copies for your district? Paying the full retail price for each one could wipe out a budget for software on one item. Negotiate a reasonable, nominal fee for additional or replacement copies. In school settings disks are sometimes lost, stepped on, or otherwise damaged. In addition, you should investigate the publisher's policy if, for some reason you didn't foresee, the program just will not work out. Do not, however, make the return option an excuse for weak evaluation practice on your part.

Finally, deal openly and honestly with publishers and expect to be treated in the same manner. Remember, they are somewhat new at this also. The copyright laws are ambiguous on computer materials and business policies and practices are just being formed. Publishers and educators must work together to establish a smooth working relationship. Tell your colleagues and educators in other districts

BEST COPY AVAILABLE

about publishers who are good to work with. Set up a network of information.

Make use of general professional resources, like reviews of software from EPIE (Educational Products Informational Services) and MicroSift. Read software reviews and ideas about instructional use of computer journals like Language Arts, English Journal, and Computers in Reading and Language Arts. But don't let others' ideas take the place of your own hands-on assessment. The ideas in this digest, along with documents like "Guidelines for Review and Evaluation of English Language Arts Software" (National Council of Teachers of English, 1984) can help you to choose the best software for use with your students.

References

Educational Products Information Exchange Institute (EPIE), 475 Riverside Drive, New York, New York 10027.

MicroSift, % Northwest Regional Educational Laboratory, 300 S.W. Sixth Avenue, Portland, Oregon 97204.

National Council of Teachers of English (NCTE), 1111 Kenyon Road, Urbana, Illinois 61801.

ERIC

A Product of the ERIC Clearinghouse on Reading and Communication Skills
1111 Kenyon Road, Urbana, Illinois 61801
1984

This publication was prepared with funding from the National Institute of Education, U.S. Department of Education, under contract no. 400-83-0025. Contractors undertaking such projects under government sponsorship are encouraged to express freely their judgment in professional and technical matters. Prior to publication, the manuscript was submitted to English language arts specialists for critical review and determination of professional competence. This publication has met such standards. Points of view or opinions, however, do not necessarily represent the official view or opinions of either the National Council of Teachers of English or the National Institute of Education.

Software Evaluation for the Teacher of the English Language Arts

Sifting good software from bad is not difficult, but it is not like selecting textbooks. If you use the same strategies, you may waste both time and money. You can flip through many textbooks and get a good idea of their worth because you are sophisticated about what you are looking for. Also, when teaching with textbooks, you are still a major part of the show—preparing, emphasizing, explaining, highlighting, filling in the gaps, and making up for weaknesses. The book will not slam shut if a student forgets to turn the pages in a certain manner. A computer program, however, can create frustration and confusion for the student. It can, in effect, "break down" if it is not thoughtfully designed. Therefore, teachers must learn selection strategies in order to become sophisticated consumers.

What Are the Initial Considerations?

Never buy software without conducting a thorough examination of it. You should look at two parts of the software: the documentation and the actual, running program. The documentation is the written explanation that comes with the computer disk. Beware of any software that does not have such documentation, as lack of documentation often means the person who produced the program didn't organize it well or didn't bother with explanatory material.

The documentation should help answer some basic questions. Are the stated objectives of the program something you need for your students? What equipment specifications does the program take? If the program needs 64,000 units of memory and your computers have only 48,000, do you want to buy additional memory chips for all your machines so that this program will run? Does the success of the program depend on color—if so, do you have color monitors? Trying to distinguish the sentences written in red from the ones in blue on a black and white screen is tiresome, if not impossible. Does the program need disk drives or cassette players? If disk drives, how many? Is one of the strengths of the program that it will print out a paper copy of the student's work? If it is and you either don't have access to a printer or have one printer for thirty students, maybe the program isn't for you.

When First Viewing the Program, What Should You Look for Concerning Organization and Structure?

Suppose you like the subject matter, accept the stated objectives, and have the equipment. It's time to examine the program for the first run-through—and this ought to be the first of several if you are going to be a competent consumer. Concentrate on three areas during the first run: Does the program carry out its stated objectives? Is it user-friendly? Does it have sound pedagogical structure?

You probably do not need help in deciding if the program carries out its stated objectives, because that's a regular part of teaching. But what about user friendliness? Is the program easy to use? Are there clear directions, or do you need technical knowledge about computers to make this program work?

In addition, examine the program for ease of operation. If you have to press the return key after your input, does it tell you to do so? Is the program menu driven? That is, do you start out with a page of choices, or do you have to start in the same place every time—reading directions at the beginning? What about a way to quit if you get bogged down or must leave? Can you get out of the program short of pulling the plug? Can you go back to see those directions you thought you didn't need in the first place, or thought that you'd remember after a quick read-through?

What Should I Know about Feedback?

It's important to look at both positive and negative feedback. These terms do not mean good and bad. Rather, they refer to the feedback given for correct and incorrect responses. Is the feedback appropriate? Too much positive feedback isn't useful. Some programs go too far, using the student's name and a string of superlatives to reward a correct response: "What a marvelous job, Charlie. You sure know your stuff." This approach is boring and often condescending. Field studies show that a simple "Okay" or "Correct" is often enough.

Moreover, does the negative feedback help point students in the right direction? Computers can handle many possible answers in many forms in a helpful way—if the programming was done by a competent programmer and a designer who knows the content. Feedback that says "Think!" is useless to students who thought they were thinking but missed the answer anyway. A more helpful feedback design is the corrective feedback paradigm (CFP). This system reruns missed questions through a questioning sequence at specified intervals to reinforce retention.

Finally, there are some programs that won't let students out until they have answered five or ten or fifteen questions correctly in a row. Some students just can't do that, and they get so frustrated by a lack of success that they quit. Keep this in mind as you evaluate a program. In addition, remember to check whether the program keeps track of the student's work, giving some type of summary and evaluation at the end.

What Pedagogical Issues Should I Consider During the First Viewing?

There are several important issues. For example, is the sequencing pedagogically logical? Does the program permit the student to interact—think, respond, wonder, predict—or does the student just read an electronic workbook? (You can buy quite a few workbooks for the cost of one piece of software, not to mention the cost of the computer.) The more interactive the program, the better. Get students involved in this evaluation and make sure that computer involvement is worthwhile for both their education and their time.

Look carefully at the language: Is it too formal, or too laden with slang? Is it so dependent on fad that it will be outdated in two years? Finally, do graphics and/or sound add anything to the program, or are they window dressing?

What Should I Look for in a Second Viewing?

If the program seems to pass the test so far—you like the objectives, it's user-friendly, and the pedagogy is good—it's time for a second round of assessment, in which you consciously make errors. This may seem hard. Teachers like to do things correctly and neatly. On this run-through, however, you'll need to force yourself to be a confused and/or slow learner.

Find out what happens if you just hit the return key with no input. Can you do it three times, get the correct answer, and page on through or do you have to make an attempt to answer? What does your program do with perceptually correct but literally incorrect answers—especially misspellings, parts of names, or abbreviations? What does the program do with a totally "off-the-wall" response—treat it as such or say, "Wow, that's interesting, Johnnie"? What happens if you hit the escape key, hit many keys simultaneously, or randomly tap the keyboard? Try it. If the program is capable of handling such things, it's said to be "bulletproof." If the program isn't designed to handle accidental or intentional problems, it's going to backfire on some students.

If you look at these aspects, you will have given the program a good preview. If you still like the program at this point, then it's time to turn the program over to some of your students. Choose at least a couple—the quick/bright and the slow/unmotivated—and get their opinions. Their actions and reactions will reveal much more about the program.

What Should I Know about Dealing with Software Publishers?

If publishers will not let you inspect programs on a trial basis, beware but remember that they are not entirely at fault. Software development, especially well-done material, costs money. If publishers send out disks for preview and you copy them and send them back, they stand to lose a great deal of money. So allow for the publisher's qualms when making arrangements for previewing. Send a letter guaranteeing that you will not copy illegally. In addition, request to see the real program rather than shortened preview disks. If they won't send just the disks, ask them to send a sales representative to do a demonstration and then try to get time for your own evaluation. Sometimes several schools can join together and invite several publishers to put on a software fair with a wider range of materials. It takes planning, but it is a way to see the material.

Another factor to check on is how to get a backup disk or multiple copies. You should be given either a backup disk or provision to make your own. Also, how does the publisher deal with your need for five, ten, fifteen or thirty copies for your district? Paying the full retail price for each one could wipe out a budget for software on one item. Negotiate a reasonable, nominal fee for additional or replacement copies. In school settings, disks are sometimes lost, stepped on, or otherwise damaged. In addition, you should investigate the publisher's policy if, for some reason you didn't foresee, the program just will not work out. Do not, however, make the return option an excuse for weak evaluation practice on your part.

Finally, deal openly and honestly with publishers and expect to be treated in the same manner. Remember, they are somewhat new at this also. The copyright laws are ambiguous on computer materials and business policies and practices are just being formed. Publishers and educators must work together to establish a smooth working relationship. Tell your colleagues and educators in other districts about publishers who are good to work with. Set up a network of information.

In addition, make use of general professional resources, like reviews of software from EPIE (Educational Products Information Exchange) and MicroSift. Read software reviews and gather ideas about the instructional use of the computer in journals like *Language Arts, English Journal,* and *Computers in Reading and Language Arts.* But don't let others' ideas take the place of your own hands-on assessment. The ideas in this digest, along with documents like "Guidelines for Review and Evaluation of English Language Arts Software" (National Council of Teachers of English 1984) can help you to choose the best software for use with your students.

Pamela Janello
University of Illinois

References

Educational Products Information Exchange Institute (EPIE), 475 Riverside Drive, New York, New York 10027.

MicroSift, c/o Northwest Regional Educational Laboratory, 300 S.W. Sixth Avenue, Portland, Oregon 97204.

National Council of Teachers of English (NCTE), 1111 Kenyon Road, Urbana, Illinois 61801.

ERIC

A Product of the ERIC Clearinghouse on Reading and Communication Skills
1111 Kenyon Road, Urbana, Illinois 61801
1984

This publication was prepared with funding from the National Institute of Education, U.S. Department of Education, under contract no. 400-83-0025. Contractors undertaking such projects under government sponsorship are encouraged to express freely their judgment in professional and technical matters. Prior to publication, the manuscript was submitted to the Editorial Board of the National Council of Teachers of English for critical review and determination of professional competence. This publication has met such standards. Points of view or opinions, however, do not necessarily represent the official view or opinions of either the National Council of Teachers of English or the National Institute of Education.

BEST COPY AVAILABLE